APPLICATION NOTE

User Guide for the Evaluation Kit ATA8510-EK1

ATAN0046

Features

- User guide for the ATA8510-EK1 evaluation kit
- Demonstrates an application with the
  - RF transceiver Atmel® ATA8515 in a base station with an Xplained PRO SAMD20 kit
  - Atmel ATA8510 RF transceiver remote sensor with an AT30TS75A temperature sensor and an optional CMM-1923 real-time clock
  - two-way RF communication

Description

This user guide describes an evaluation kit for industrial RF applications having the following components:

- A base station using an
  - Xplained PRO SAMD20 evaluation kit
  - Xplained PRO OLED1 extension board
  - Xplained PRO ATA8510/15 extension board
- Remote temperature sensor with Atmel ATA8510 running a Flash application

The Atmel ATA5815 is used as an RF transceiver in the base station and runs in polling mode to detect data telegrams and displays the received message on the OLED. The received message is also streamed to a PC terminal program using a virtual COM port connection.

The remote sensor includes an AT30TS75A temperature sensor device and a CMM-1923 real-time-clock device for waking up the Atmel ATA8510 RF transceiver. The Flash application reads the temperature data from the sensor device using a TWI bus protocol implemented in software and broadcasts the temperature data via the RF link.

The RF application uses a 2-way communication, i.e. the transmitted RF telegrams are acknowledged from the receiver.

The RF link operates on channel 433.92MHz at an 8kBit/s data rate using FSK modulation with Manchester encoding.

References

[1] Atmel® ATA8510/ATA8515 datasheet

[2] Atmel AT30TS75A datasheet

[3] C-MAX CMM-1923-V1.0 datasheet

[4] http://www.atmel.com

[5] http://www.iar.com

[6] Atmel ATAN0096 - ATA8510 Programmers Guide

[7] Atmel ATAN0035 - ATA583x and ATA578x Configuration Tool Guide and software

[8] Atmel ATAN0036 - ATA583x and ATA578x Flash Application Development

[9] Atmel ATA8510/ATA8515 User Manual

[10] ATA8510-EK1_Tool_Pack_V1.0.zip

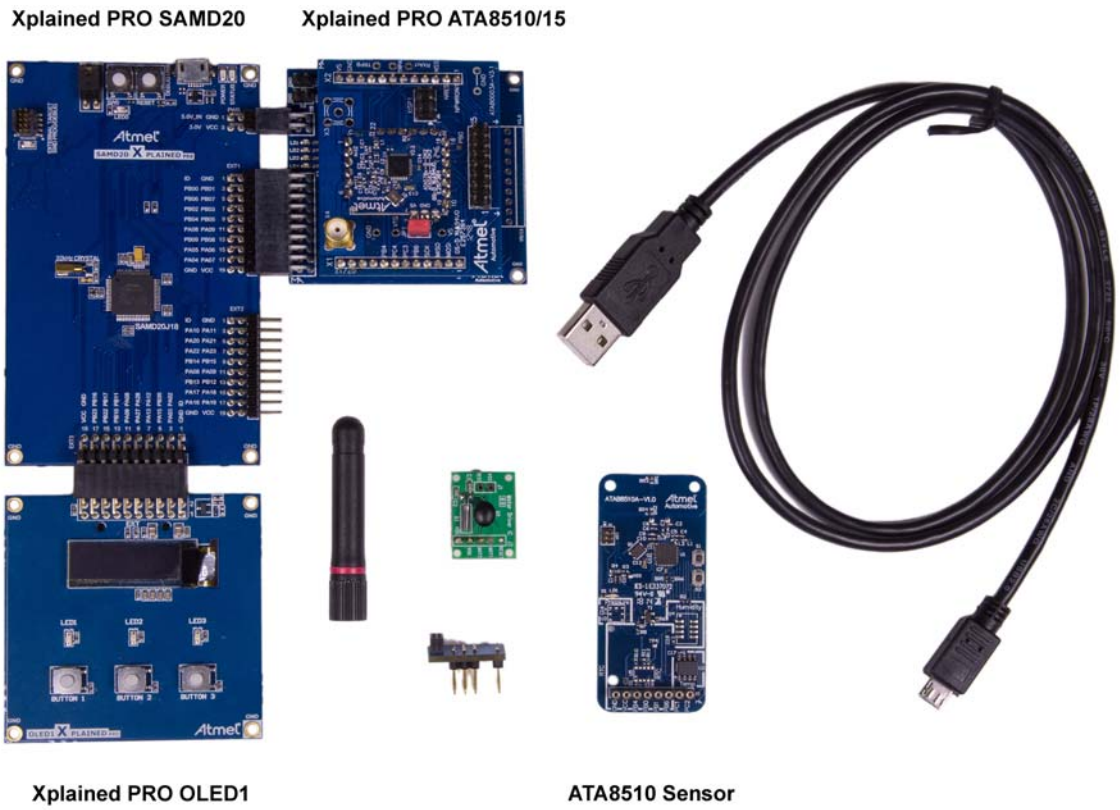
1. Getting Started

Figure 1-1 shows the components which are included in the evaluation kit. The kit includes

- A base station with
  - Xplained PRO SAMD20 evaluation board
  - Xplained PRO OLED1 extension board
  - Xplained PRO ATA8510/15 extension board
- A remote temperature sensor with Atmel® ATA8510 [1], AT30TS75A [2], and an optional CMM-1923 [3] devices
- A mini USB cable
- A mini ISP adapter
- A 433MHz whip antenna

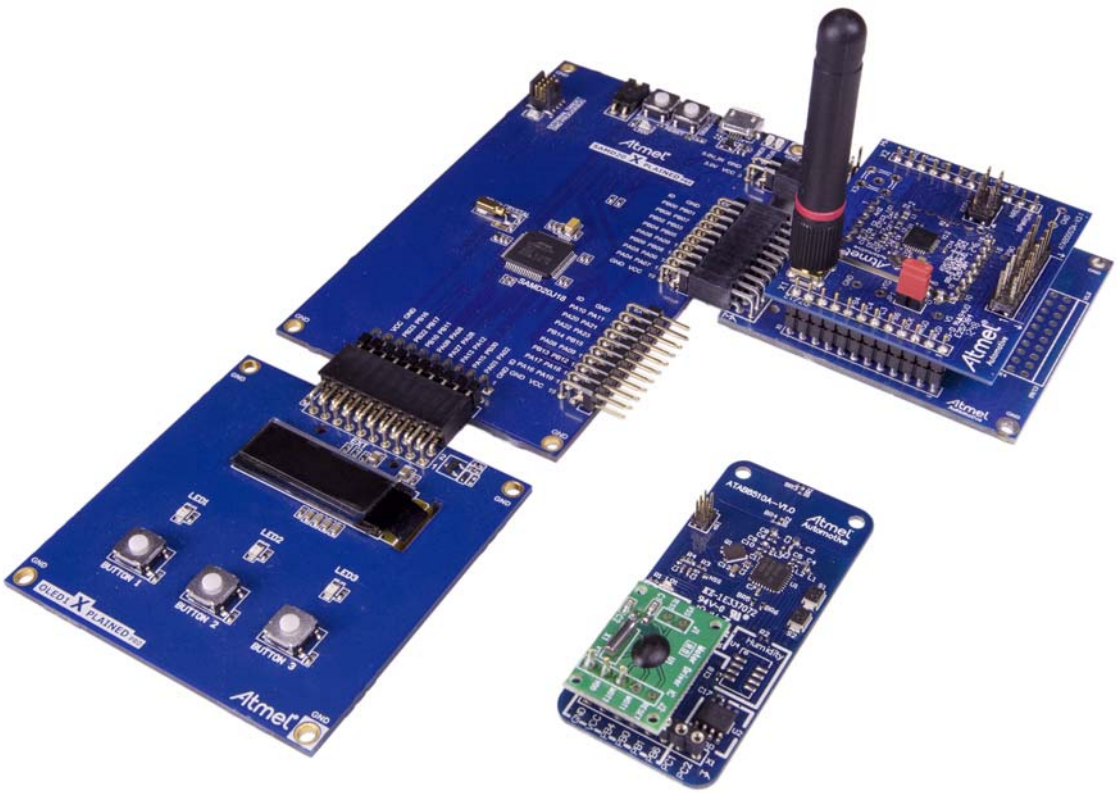
The remote sensor requires a CR2032 coin cell battery not included in the kit.

Figure 1-1. ATA8510-EK1 Kit Components



The Xplained PRO OLED1 board is connected to the EXT3 extension header and the Xplained PRO ATA8510/15 board is connected to the EXT1 extension header of the Xplained PRO SAMD20 board as shown in Figure 1-2. The 433MHz whip antenna is mounted on the X4 connector of the Xplained PRO ATA8510/15 board. These boards are powered using the USB cable connected to the debug USB connector of the Xplained PRO SAMD20 board. The USB cable can be connected to a 5V/500mA USB power supply for stand-alone operation. When using the virtual COM port, the USB cable is connected to a PC's USB port, which requires previous installation of Atmel Studio 6 IDE [4]. This installation includes all required USB drivers for operation of the Xplained PRO SAMD20 board.

Figure 1-2. ATA8510-EK1 Kit



Atmel® Studio 6 can be downloaded from [4] with additional user instructions and tools. Documentation for the Xplained PRO SAMD20 and Xplained PRO OLED1 are also available from [4]. Documentation for other components in the kit and the software is included in the tool pack zip folder [10] available from your Atmel sales representative.

The OLED displays a welcome screen after power-up

```
ATA8510-EK1 Demo Kit
(c)Atmel 2015 V1.1
Free RTOS (FW2.1)
Waiting for RF signal...
```

and waits for an RF telegram from the remote sensor. After installing the CR2032 coin cell into the remote sensor (be sure to observe correct polarity), the sensor immediately starts sending RF telegrams at 2s intervals when the optional RTC module is installed. The OLED shows the temperature value from the sensor with a resolution of 0.5°C. The time interval between the RF signal reception in seconds and the signal strength rssi on the base station and RSSI on the sensor is also shown:

```
 ____________________
| dt=2s, rssi=221   |
|      RSSI = 128   |
|      T= 22.5°C    |
|___________________|
```

When removing the CMM-1923 real-time clock, the transmission of an RF telegram can be initiated by pressing the S2 button (for more information, see Figure 3-2 on page 9). The display can be switched to different modes via the buttons 1-3 on the Xplained PRO OLED1 extension board (for more information, see Section 4.2 “Base Station Application for SAMD20 MCU Using freeRTOS” on page 12).

The tool pack zip folder [10] contains the documentation and software for the kit. Extract the folder into a directory on the PC to get the following structure:

..\Application_Notes	Documentation for the kit and the application notes for the devices
..\Documentation	Device datasheets
..\Hardware	Hardware documentation of the boards
..\Software	Kit software and tool support extensions
..\Software\Programming_Files	Device programming files in *.hex format
..\Software\IDE_Support_Files	Tool support files for IAR Workbench and Atmel Studio 6
..\Software\base	Software project and sources for the Xplained PRO kit
..\Software\base\EEPROM	Configuration file for the Atmel ATA8515 device of the base station
..\Software\base\SAMD20	Atmel Studio 6 project directory
..\Software\sensor	Software project and sources for the remote sensor
..\Software\sensor\EEPROM	Configuration file for the Atmel ATA8510 device of the remote sensor
..\Software\sensor\ATA5831_FLASH_IAR_2.32.0	IAR Embedded Workbench project directory

2. Data Protocol and Signal Timing

The remote sensor is sending a RF data telegram every 2s or when the button S2 is pressed with the temperature data. The base station receiver is polling for this data telegram on RF channel 433.92MHz in regular intervals of 5ms. The RF transmission and data protocol of the remote sensor and the base station is using the following settings:

- Transmission order is MSB first for preamble and data section
- RF channel: 433.92MHz
- Data rate: 8kBit/s with FSK modulation and Manchester encoding
- Deviation: ±8kHz

Each data packet uses 2 bits before and after the data telegram to force a Manchester code violation, which is detected by the receiver to separate the data packets und to restart the synchronisation procedure of the receiver.

2.1 Data Protocol for Remote Sensor Telegram

The remote sensor data protocol parameters are shown in Table 2-1. The Manchester encoding is using a low-to-high transition for a '0' bit and a high-to-low transition for a '1' bit.

Table 2-1. Data Protocol

Protocol Item	# of Data Bits	Value	Description
Preamble	55	'1'	Pre-burst for wake-up and synchronisation
Start bit	1	'0'	Start bit to indicate data payload
Data payload	8	0x14	ID for no valid sensor data
	8	0x19	ID for low battery voltage (V_{CC} < 2.2V)
	8	0x64	ID for temperature data (–55 to +125°C)
	16	Signed integer	Temperature value (0.1°C resolution)
	8	Checksum	Checksum of ID and data payload as 2th complement
Total	88		Data telegram with Ttx = 11ms at 8kBit/s

The acknowledge protocol send by the remote sensor has the following data:

Table 2-2. Acknowledge Protocol

Protocol Item	# of Data Bits	Value	Description
Preamble	55	'1'	Pre-burst for wake-up and synchronisation
Start bit	1	'0'	Start bit to indicate data payload
Data payload	8	0x60	ID for RSSI data
	16	Integer	RSSI value [0-255]
	8	Checksum	Checksum of ID and data payload as 2th complement
Total	88		Data telegram with Ttx = 11ms at 8kBit/s

2.2 Data protocol for Base Station Acknowledge

The base station acknowledge protocol parameters are shown in Table 2-3. The Manchester encoding is using a low-to-high transition for a '0' bit and a high-to-low transition for a '1' bit.

Table 2-3. Acknowledge Protocol

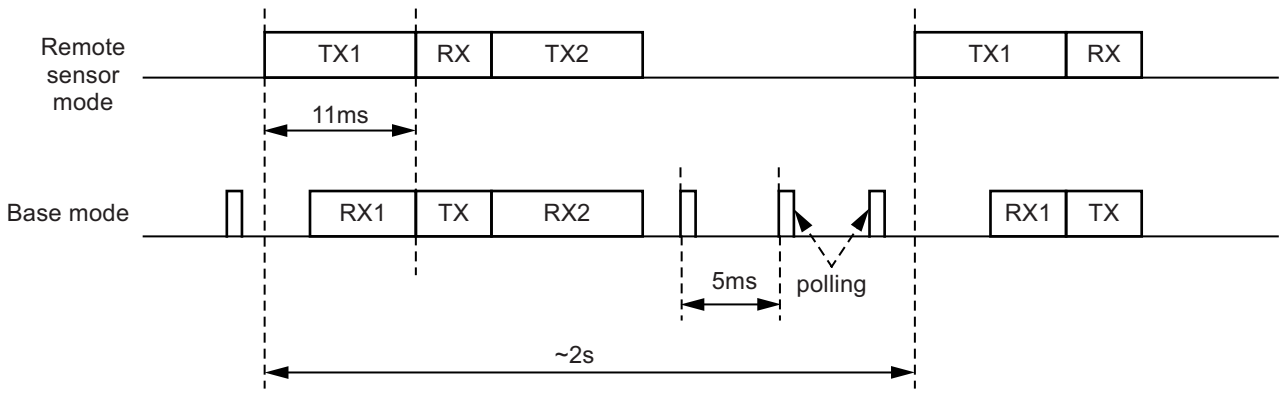
Protocol Item	# of Data Bits	Value	Description
Preamble	55	'1'	Pre-burst for wake-up and synchronisation
Start bit	1	'0'	Start bit to indicate data payload
Data payload	8	0x60	ID for RSSI data
	8	0x60	Repeated
Total	72		Data telegram with Ttx = 9ms at 8kBit/s

2.3 Signal Timing

The signal timing of the data transfer is shown in Figure 2-1.

1. the remote sensor transmits a data telegram with duration of 11ms every 2s or when button S2 is pressed (TX1). The pre-amble duration is 7ms while the data payload duration is 4ms. The remote sensor switches then into receive mode with a timeout period of 20ms (RX).
2. the base station is polling for a valid RF signal which matches the RF channel, the data rate, the modulation scheme and the correct data encoding every 5ms. This polling interval is chosen to match with the length of the pre-amble. When a valid telegram is received (RX1) the base station immediately acknowledges this telegram (TX) and waits for an acknowledgement frame from the remote sensor with the remote sensor signal strength data and a timeout of 90ms (RX2).
3. The remote sensor transmits the acknowledgement frame with the RSSI signal strength (TX2) and switches then into OFF mode.
4. The base station reads the acknowledgement frame with the RSSI data (RX2), displays the result and switches into polling mode again.

Figure 2-1. Signal Timing



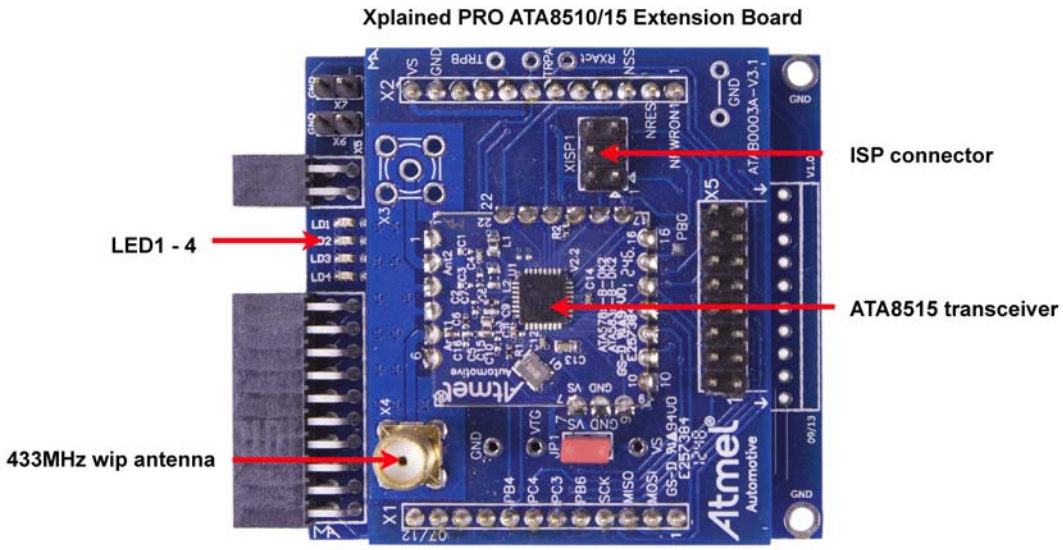
3. Hardware Description

The description and documentation of the Xplained PRO SAMD20 board and the Xplained PRO OLED1 board are available within Atmel Studio 6 or on the Atmel® website [4]. The hardware description of the Xplained PRO ATA8510/15 extension board and the remote sensor are included in the tool pack zip folder [10]. The base station is powered by the USB cable whereas the remote sensor uses a CR2032 coin cell battery.

3.1 Base Station Transceiver

The Atmel ATA8515 base station transceiver is mounted on an adapter board as shown in Figure 3-1. This adapter board is plugged onto the Xplained PRO ATA57810/15 extension board which includes the LEDs 1-4. The adapter board has a programming adapter XISP1 which is used for programming the EEPROM configuration data into the device. This extension board is connected to the EXT1 and PWR header of the Xplained PRO SAMD20 board. The extension board has two additional connectors, X6 for supplying the kit with 5V and X7 to access the kit's internal 3V supply voltage.

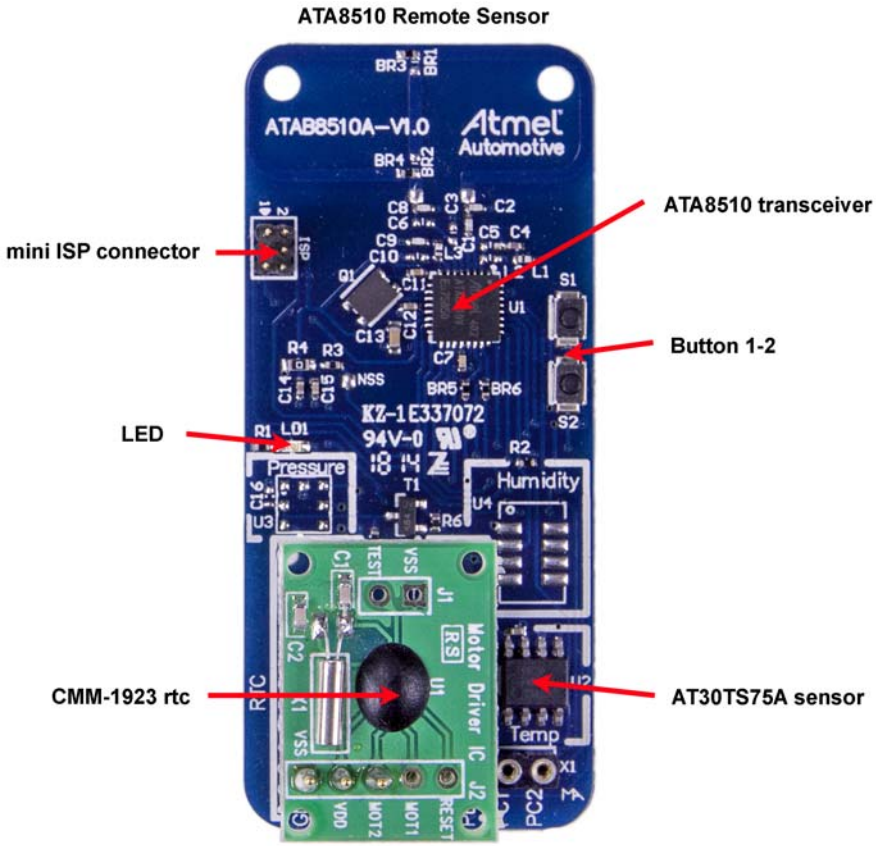
Figure 3-1. Base Station Transceiver Board



3.2 Remote Sensor Transceiver

The Atmel ATA8510 remote sensor transceiver is placed on a key fob board as shown in Figure 3-2. This remote sensor is powered by a CR2032 coin cell battery (not included) mounted on the bottom of the board. The board contains the real-time clock CMM-1923 [3] attached to the connector X1 and the AT30TS75A [2] temperature sensor. The S1 and S2 buttons are placed on the board with the S1 button not used in this application and the S2 button used for initiating data telegram transmission. The LED flashes during a temperature measurement before transmitting the RF data telegram. The mini ISP header is used for Flash and EEPROM programming and for Flash application debugging of the Atmel ATA8510 (see [8] for more details about Flash application development).

Figure 3-2. Remote Sensor Transceiver Board



The current consumption profile over time is shown in Figure 3-3. During the temperature measurement period of 40ms, the LED is switched on in parallel, increasing current consumption. Figure 3-3 shows current consumption for the measurement period with and without LED. The RF telegram transmission takes about 11ms at a power level of 6dBm, resulting in current consumption of about 9.4mA. In OFF mode the temperature sensor is switched off and the resulting current is indicated by the Atmel ATA8510 transceiver and the CMM-1923 RTC current consumption. This OFF mode current is highly dependent on ambient temperature shown in the datasheet [1] and [3]. Figure 3-3 indicates current consumption measured at a room temperature of 24°C.

Figure 3-3. Remote Sensor Current Consumption at 24°C

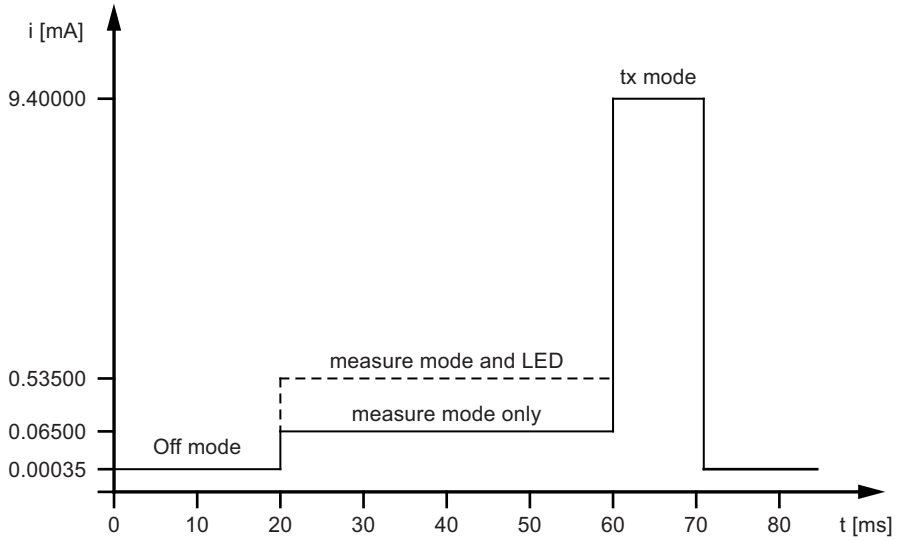


Table 3-1 summarizes battery life for a CR2032 and CR2450 coin cell without using the LED during temperature measurement at 20°C. It is assumed that the battery voltage drops from the initial value of 3V down to 2V by the end of battery service. The self discharge is defined by the manufacturer with typically 1% per year at an ambient temperature of 20°C. The battery life for this application primarily depends on the measurement interval Ts, which defines the duration of the OFF mode whereas measurement and transmit mode have fixed durations.

Table 3-1. Battery Life at 20°C

Interval Ts	Mean Current	Battery Lifetime (Days/Years)			
(s)	(mA)	CR2032 (230mAh)		CR2450 (560mAh)	
		Days	Years	Days	Years
2	0.053	180	0.5	437	1.2
5	0.022	445	1.2	1083	3.0
10	0.011	875	2.4	2131	5.8
20	0.006	1696	4.6	4130	11.3
50	0.002	3880	10.6	9448	25.9

4. Software Description

The demo application uses the following programs:

1. A Flash application with EEPROM settings for the Atmel® ATA8510 remote sensor
2. An SAMD20 application with EEPROM settings for the Atmel ATA5815 receiver

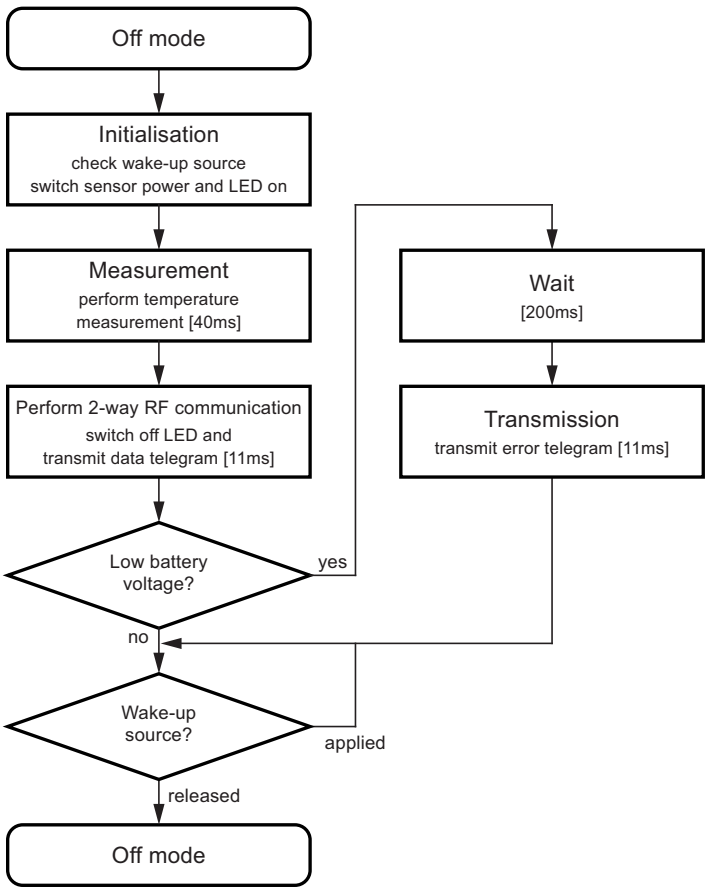
The following section describes each program flow. The IAR embedded workbench for AVR® [5] is used together with the JTAGICE3 debug tool for Flash application development of the remote sensor. Atmel Studio 6 [4] is used together with the debug tool included on the Xplained PRO SAMD20 board for SAMD20 application development.

The source code for all applications is available within the tool pack distribution [10].

4.1 Flash Application for the ATA8510 Remote Sensor

The program flow for the remote sensor application is illustrated in Figure 4-1. The initial state of the device is the OFF mode. The real-time clock of the remote sensor wakes up the transceiver every 2s and switches the device to active mode. The temperature sensor is powered up and read out followed by the transmission of the data telegram. During transmission of this data telegram the supply voltage level is checked and if this is below 2.2V, an error message is transmitted after a delay of 200ms. The device then enters the OFF mode again until the next wake-up by the RTC (or by pressing the S2 button).

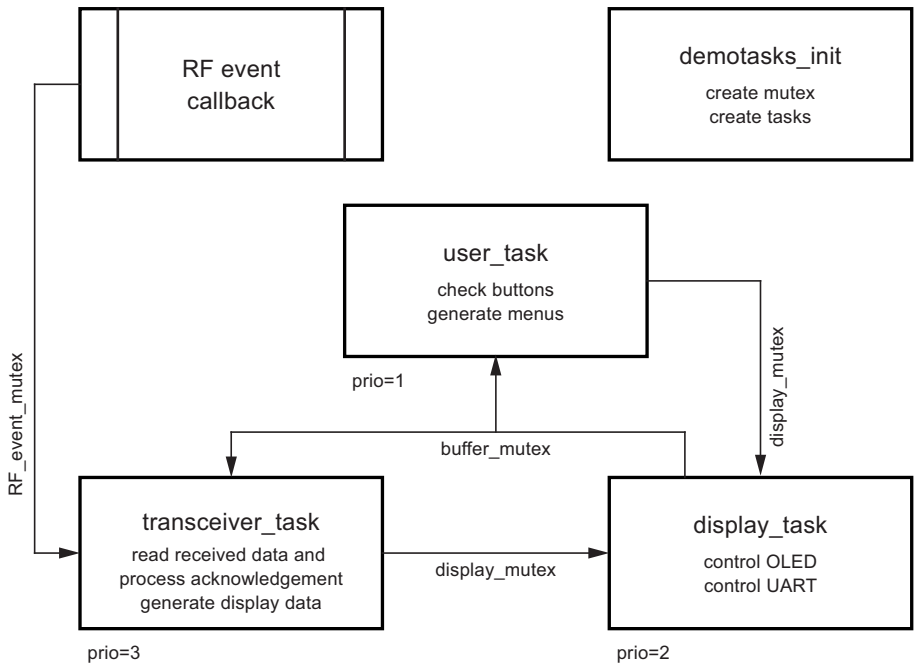
Figure 4-1. Flash Application for the Atmel ATA8510 Remote Sensor



4.2 Base Station Application for SAMD20 MCU Using freeRTOS

The program flow for the SAMD20 application is illustrated in Figure 4-2. This application uses freeRTOS (http://www.atmel.com) as an operating system for the device and includes a total of three tasks with different priority levels. The tasks and the mutex objects are instantiated and initialized before starting the scheduler in the demotask_init() function. After initializing the hardware and peripherals and before entering an infinite loop, the main() function (not shown in Figure 4-2) calls this demotask_init(). The RF event callback function indicates to the transceiver_task() the occurrence of an RF transceiver event using the RF_event_mutex object. The user_task() checks for button presses and correspondingly switches the display. The display_task() shows the text strings generated by the transceiver_task() or the user_task(). The display activity is controlled by the buffer_mutex and display_mutex mutexes. The buffer_mutex controls access to the display buffer and the display_mutex controls access to the OLED and the UART.

Figure 4-2. Flash Application for SAMD20 MCU



The task description and the text strings generated by the user_task() and the transceiver_task() are as follows:

- user_task(): handles the user entry with buttons1-3 and creates the start page and menus to be displayed:

  Start page: Shows the welcome message with software and transceiver firmware version information and awaits a remote sensor signal.

```
ATA8510-EK1 Demo Kit
  (c)Atmel 2015 V1.1
  Free RTOS (FW2.1)
Waiting for RF signal...
```

  Button 1 menu: Shows the statistics with the number of valid data telegrams which have passed the checks, the number of error telegrams which have failed during checking and the total number of telegrams including false detection.

```
Receiver statistics:
valid# 3062
error# 15
total# 3099
```

  Button 2 menu: Shows parameters used in the RF telegram.

```
RF-channel 433.92MHz
Data rate 8kBit/s
FSK deviation ±8kHz
Manchester coding
```

Button 3 menu: Shows parameters used for the virtual COM port. These parameters have to be set in the PC terminal application. To receive the text strings.

```
COM port settings
baudrate 9.6Kbaud
8 data + 1 stop bit
no parity, no handshake
```

- transceiver_task(): Reads the received data from the transceiver device, checks for data consistency and creates a text string for display:

  The measurement display shows the time interval dt between reception of two consecutive RF telegrams together with the RSSI value for the RF signal strength. The temperature at the sensor is shown with a resolution of 0.5°C.

```
 ____________________
| dt=2s, rssi=221   |
|      RSSI = 128   |
|      T= 22.5°C    |
|___________________|
```

  Or:

  This message is shown when the received data telegram is corrupted or when the temperature sensor is broken or not present.

```
!!!!!!!!!!!!!!!!!!!!!
Sensor error:
No valid sensor data!
!!!!!!!!!!!!!!!!!!!!!
```

  Or:

  This message is shown when the sensor battery voltage is less than 2.2V and needs to be replaced.

```
!!!!!!!!!!!!!!!!!!!!!
Sensor error:
Low battery voltage!
!!!!!!!!!!!!!!!!!!!!!
```

  Or:

  This message is shown when the RF receiver detects a signal on another RF channel (this must be enabled in the EEPROM configuration of the receiver).

```
:::::::::::::::::::::
RF channel error:
Wrong RF telegram!
:::::::::::::::::::::
```

- display_task(): Prints the text string from the user_task() or transceiver_task() on the OLED and sends the text string via the virtual COM port to the PC terminal window.

The status of the tasks is displayed on LED0 of the Xplained PRO SAMD20 board and on LED1-3 of the Xplained PRO OLED extension board in parallel to the LEDs on the Xplained PRO ATA8510/15 extension board as listed in Table 4-1.

Table 4-1. LED Functions

Xplained PRO SAMD20	Xplained PRO OLED1	Xplained PRO ATA8510/15	Description
LED0	--	LED1	Indicates an active RF event
--	LED1	LED2	Indicates an active display_task()
--	LED2	LED3	Indicates an active transceiver_task()
--	LED3	LED4	Indicates an active user_task()

The buttons of the OLED extension board have the function as described above in the user_task() while SW0 on the Xplained PRO SAMD20 board has no function:

BUTTON1	Shows the receiver statistics
BUTTON2	Displays RF parameter settings
BUTTON3	Displays COM port settings

4.3 Software Development

Atmel Studio 6 is required for base station software development and can be downloaded from the Atmel website [4]. In addition, the tool pack [10] with software and documentation is required. IAR Embedded Workbench for AVR [5] and the JTAGICE3 AVR® debug tool [4] are required for remote sensor software development.

4.3.1 Xplained PRO SAMD20 Base Station

The project file for base station software development is located in the extracted folder in the subdirectory

..\Software\base\SAMD20	Atmel Studio 6 project directory

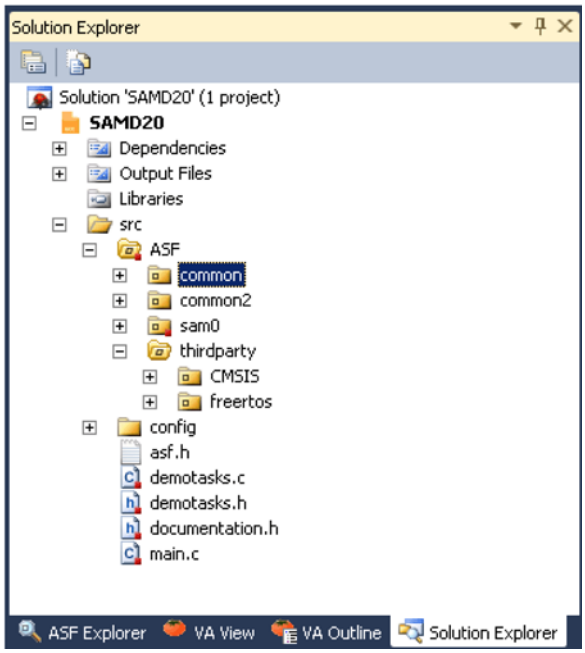
After installation of Atmel Studio 6, the USB driver is automatically installed when connecting the Xplained Pro SAMD20 board for the first time. When opening Atmel Studio 6, you will see the landing page of this kit were you can obtain the documentation and sample projects, i.e., including FreeRTOS examples. The OLED display shows the welcome screen described in the user_task() and waits for a remote signal from the remote sensor. After attaching the battery to the remote sensor, the RF telegram is immediately transmitted with an update of the temperature, RF telegram interval and RSSI signal strength parameter at 2s intervals. You can also view the received information by opening a PC terminal application and selecting the COM port of the Xplained PRO board with a baud rate of 9.6Kbaud, 8 data bit, 1 stop bit no parity and no handshake selected.

To start evaluating the software, you can select the following files from the solution window inside Atmel Studio 6 (see Figure 4-3):

main.c	Program start with initialization and activation of the RTOS tasks
demotasks.c	FreeRTOS implementation of the demo software

The implementation can be analyzed and debugged together with the program flow shown in Figure 4-2 on page 12. Additional documentation regarding the other ASF functions and the FreeRTOS can be obtained within Atmel Studio 6 Help and from http://www.freertos.org.

Figure 4-3. Base Software



Debugging of the application is performed within Atmel Studio 6 together with the Xplained PRO SAMD20 board. This board includes an embedded debugger which is controlled by Atmel Studio 6.

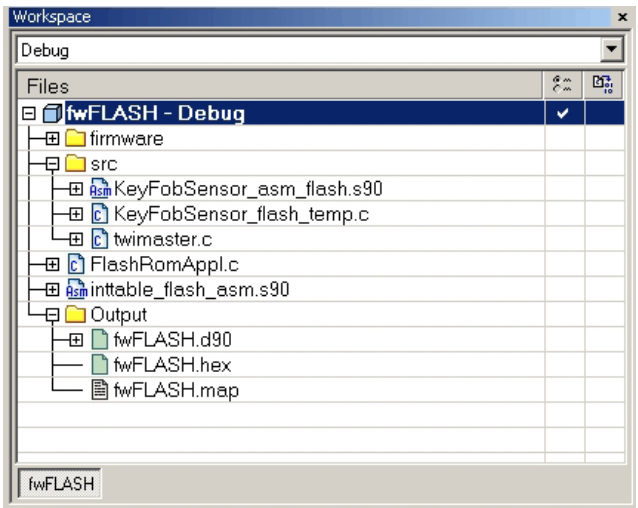
4.3.2 Remote Sensor

IAR Embedded Workbench for AVR is required for remote sensor software development [5]. The project file is located in the extracted subfolder:

..\Software\sensor\ATA5831_FLASH_IAR_2.32.0 IAR Embedded Workbench project directory

After opening the IAR project Remote_sensor.eww file, the following workspace window is available (see Figure 4-4).

Figure 4-4. Remote Sensor Software



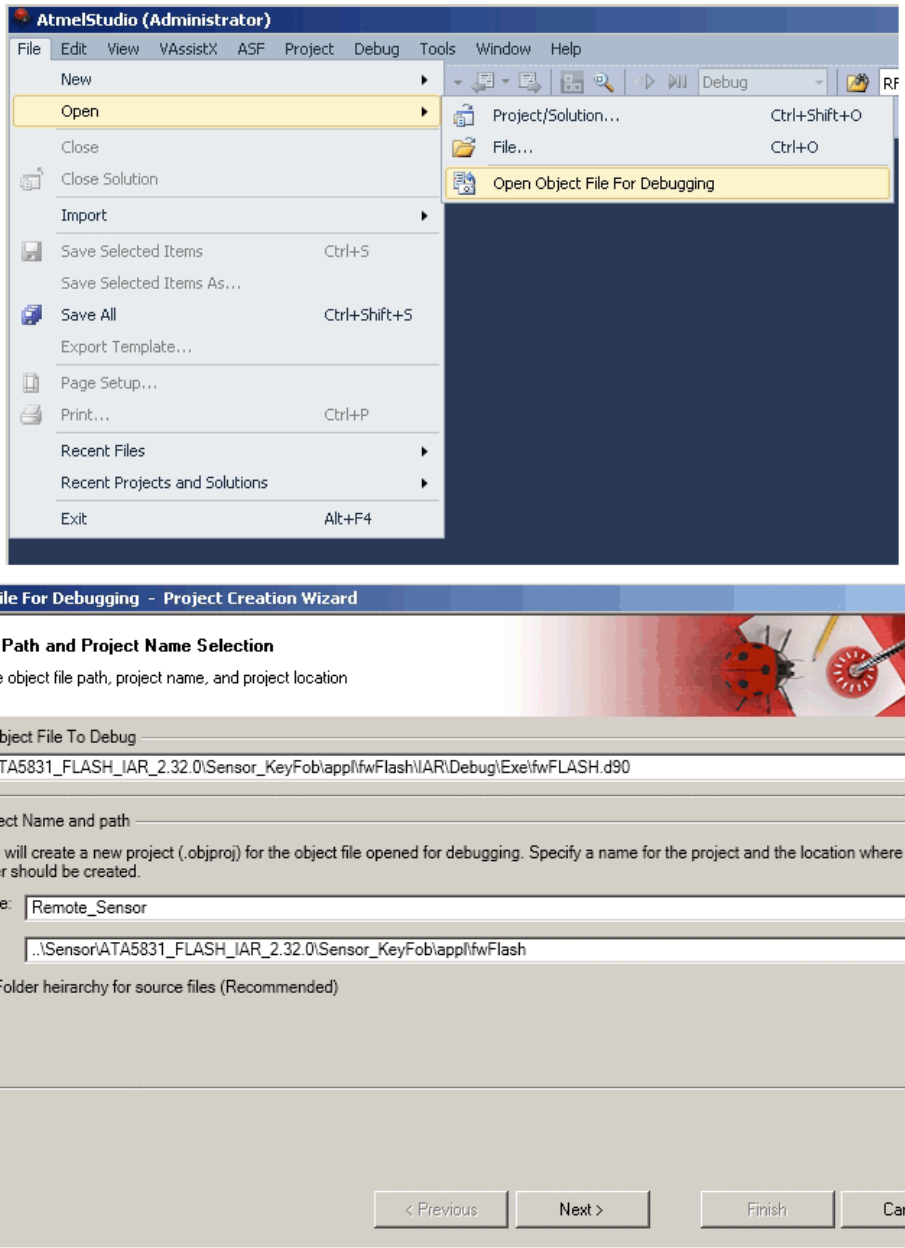
Select the following files from the workspace window to start evaluating the software:

FlashRomAppl.c	Main() program start with initialization and activation of the application loop
KeyFobSensor_flash_temp.c	Application functions called within the main() function

The implementation can be analyzed, compiled, and linked together with the program flow shown in Figure 4-1 on page 11.

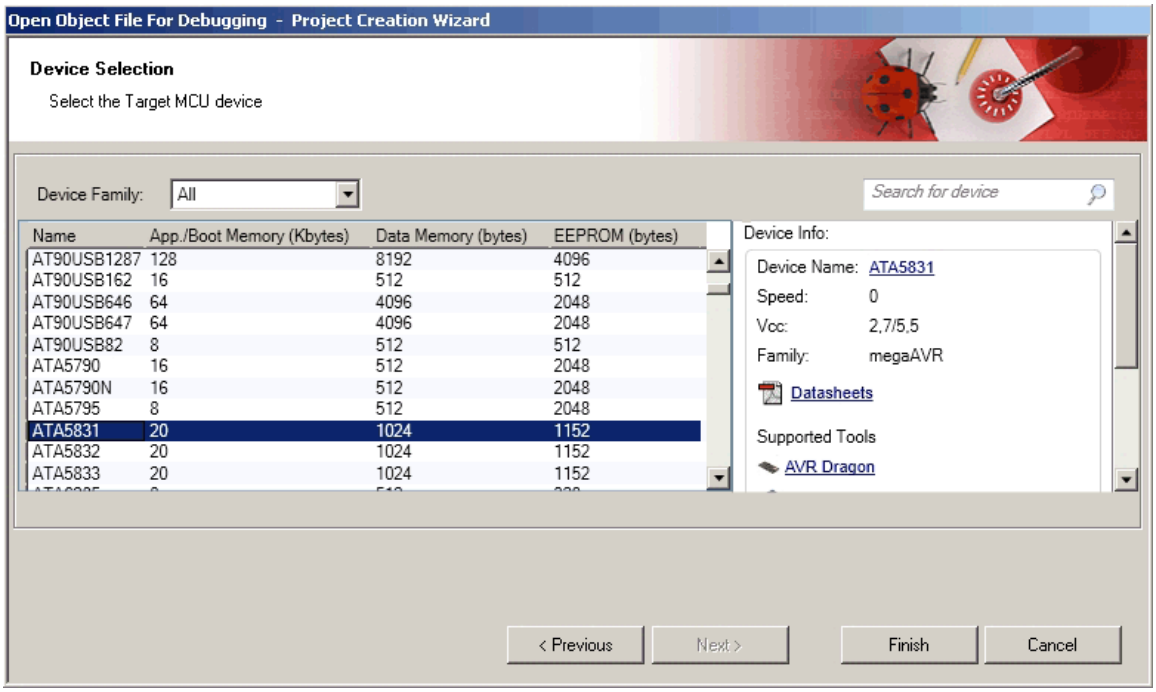
Debugging of the application is currently not supported within IAR Embedded Workbench using the AVR® debug tool JTAGICE3. Atmel Studio 6 has to be used instead. Once the program is compiled and linked within IAR Workbench, the file fwFLASH.d90 has to be opened within Atmel Studio 6 as shown in Figure 4-5. To select the right directory paths use the <...> buttons to locate the *.d90 file and for the 'Location'.

Figure 4-5. Debugging the Remote Sensor Software


AtmelStudio (Administrator)
File Edit View VAssistX ASF Project Debug Tools Window Help
New
Open
Close
Close Solution
Import
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Page Setup...
Print... Ctrl+P
Recent Files
Recent Projects and Solutions
Exit Alt+F4
Debug
Project/Solution... Ctrl+Shift+O
File... Ctrl+O
Open Object File For Debugging
Open Object File For Debugging - Project Creation Wizard
Object File Path and Project Name Selection
Specify the object file path, project name, and project location
Select the Object File To Debug
..\Sensor\ATA5831_FLASH_IAR_2.32.0\Sensor_KeyFob\appl\fwFlash\IAR\Debug\Exe\fwFLASH.d90
Specify Project Name and path
AtmelStudio will create a new project (.objproj) for the object file opened for debugging. Specify a name for the project and the location where the project folder should be created.
Project Name: Remote_Sensor
Location: ..\Sensor\ATA5831_FLASH_IAR_2.32.0\Sensor_KeyFob\appl\fwFlash
Maintain Folder heirarchy for source files (Recommended)
< Previous
Next >
Finish
Cancel

After selecting Next>, the device has to be selected. The Atmel® ATA8510 device is not currently listed and the ATA5831 device (the similar automotive device) has to be selected instead as shown in Figure 4-6. Selecting “Finish” shows the project window and after selecting the debug tool, the debugging can be performed much the way you would when using the base software.

Figure 4-6. Device Selection for Debugging



The Atmel ATA8510 device uses the debugWire interface for the debug connection (described in the user manual [9]). This connection uses the reset line on pin PC0 of the device and must be enabled within ISP programming mode. If debugWire mode is enabled, the ISP mode is not available and vice versa. Select “Disable debugWire connection” within the “Debug” menu to return to ISP mode. The 6-pin ISP connector is used as a connection to the debugger and it is recommended to connect only the V_{CC}, GND and reset signal when debugging, leaving the SPI signals open for the application. This is important because the TWI driver software uses the PB1 pin as an SCL line shared with the ISP and SPI peripheral. Additional information on how to perform development and debugging with the Atmel ATA8510 device is provided in [6] and [8]. This application note describes development for the Atmel ATA5831 device which, in terms of Flash development, is similar to the Atmel ATA8510 device.

4.3.3 Re-programming of the Devices

The tool pack [10] includes the original programming files to re-program all devices in the kit.

1. For re-programming of the SAMD20 device select the device ATSAMD20J18 in the device programming tool of Atmel Studio 6 and the file SAMD20_Flash.hex for the flash memory.
2. For re-programming of the Atmel ATA8515 device on the XplainedPRO ATA8510/15 extension board disconnect the board and apply an external 3V power supply to the connector X7. Connect pin 5 of connector X3 to GND (to prevent the device from entering the OFF mode) and select the device ATA5833 in the device programming tool and ensure that the ISP clock is set < 100kHz. Select the file Base_Station_EEPROM.hex for the EEPROM memory. For the fuse settings the fuses SPIEN and EESAVE should be set.
3. For re-programming of the Atmel ATA8510 device of the remote sensor attach the mini-ISP connector to the connector ISP and the debug tool and insert the CR2032 battery or apply an external 3V power supply to the mini ISP connector. Remove the CMM1923 rtc module and connect pin VCC to PB4 on connector X1 (to prevent the device from entering the OFF mode). Select the device ATA5831 in the device programming tool and ensure that the ISP clock is set < 100kHz. Select the file Remote_Sensor_Flash.hex for the flash memory and the file Remote_Sensor_EEPROM.hex for the EEPROM memory. For the fuse settings the fuses SPIEN and EESAVE should be set.

4.4 EEPROM Data Settings

The EEPROM of the Atmel ATA8510 and ATA8515 device includes the configuration of the device and the settings for the RF protocol. This EEPROM must be programmed before running application software either as an embedded Flash application or as a host application using the SPI communication link. The Atmel ATA5831 device has to be selected instead of the Atmel ATA8510 or ATA8515 within the programming dialog of Atmel Studio 6. Make sure that the ISP signals are available for the programmer only and that the device is not in OFF mode by pulling an NPWRONx pin to GND level, i.e., connect the PC1 pin to GND.

A Java GUI tool is used for generating the EEPROM programming file in HEX format (see [7] for the tool software and user guide, which is not included in the tool pack [10]). This tool can save all settings in an xml file and handling is described in the user guide [7] and the settings in the user manual [9].

The tool pack includes two subdirectories which contains the xml and HEX files for the base and the remote sensor:

..\Software\base\EEPROM	Configuration file for the Atmel ATA8515 base station device
..\Software\sensor\EEPROM	Configuration file for the Atmel ATA8510 remote sensor device

Only service 0 with channel 0 is used for the RF telegram. All other service settings can be ignored. For the base station the polling is defined in the polling tab of the GUI.

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**

© 2015 Atmel Corporation. / Rev.: 9343C–INDCO–07/15

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, AVR®, and others are registered trademarks or trademarks of Atmel Corporation in U.S. and other countries. Other terms and product names may be trademarks of others.

DISCLAIMER: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

SAFETY-CRITICAL, MILITARY, AND AUTOMOTIVE APPLICATIONS DISCLAIMER: Atmel products are not designed for and will not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (“Safety-Critical Applications”) without an Atmel officer's specific written consent. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Atmel products are not designed nor intended for use in military or aerospace applications or environments unless specifically designated by Atmel as military-grade. Atmel products are not designed nor intended for use in automotive applications unless specifically designated by Atmel as automotive-grade.